An SGML Environment for STEP

Lisa Phillips

Joshua Lubell

National Institute of Standards and Technology
A127 Metrology Building
Gaithersburg MD 20899

Internet: phillips@cme.nist.gov, lubell@cme.nist.gov

November 1994

1 Introduction

As part of the effort towards the advancement of the evolving Standard for the Exchange of Product Model Data (STEP), the National Institute of Standards and Technology (NIST) will be using the Standard Generalized Markup Language (SGML), an international standard for document representation and exchange of text data. SGML will be used to facilitate intelligent, structure-based access to STEP documents. Because STEP is an evolving standard, consisting of increasingly large and complex documents, the development and management of STEP documents has been both difficult and time-consuming. An SGML environment for STEP is expected to alleviate these problems by providing a neutral, STEP intelligent environment in which to develop quality documents. The development of the SGML environment for STEP will be part of a larger project called the Application Protocol Development Environment (APDE), a project charged with improving and expediting the development of STEP Application Protocols.

This paper will provide an in-depth discussion of the components of a proposed SGML environment for STEP. This section introduces the SGML environment for STEP and provides a definition of SGML and its applicability to STEP. Section 2 provides a background discussion of the aforementioned APDE project and its relationship to the SGML environment for STEP. Section 3 provides additional basic concepts in SGML. Section 4 provides further discussion on the benefits of using SGML and the rationale for choosing SGML for STEP. Section 5 provides a detailed overview of the components of an SGML environment for STEP. Finally, section 6 concludes with an overview of the lessons learned, current status, and the near-term and long-term plans for the STEP SGML environment.

Trade names and company products are mentioned in the text in order to adequately specify experimental procedures and equipment used. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the products are necessarily the best available for the purpose.

2 The Application Protocol Development Environment

The goal of the APDE is to provide an automated environment to facilitate the development of STEP Application Protocols and to improve their quality [Clark93b]. An Application Protocol (AP) is the specification of product data in a particular application area. An AP is typically large (several hundred pages of documentation) and consists of various components including textual descriptions of data and data models. The APDE will provide a tightly integrated collection of NIST-developed and commercially available software to allow AP developers to interactively create components of an AP and to store and retrieve APs, Integrated Resources[1], and other STEP-related documents and data models. The integrated tools will enable reuse of Integrated Resources and allow users to perform the various functions of AP development in a single, cohesive environment.

The core of the APDE will be an information base of STEP documents and data called the Application Protocol Information Base (APIB). The APIB will be located at NIST and will provide remote access database services. The APIB will initially coexist with the STEP On-Line Information Service (SOLIS), a public archive of documents and files administered at NIST for members of the STEP community [Rinaudot]. The APIB will expand upon SOLIS by providing direct multi-user access to documents and structure-based browsing and querying capabilities, a large part of which will be facilitated by the use of SGML. Also, documents stored in the APIB will be dynamically accessible to and from the other components of the APDE.

NIST also maintains a remote execution facility called the EXPRESS Server where users can run NIST-developed software via e-mail [Libes]. The APIB will also be accessible remotely through the EXPRESS Server and will support database query or retrieval operations, configuration management functions, and access control of STEP-related documents and data.

A high level view of the APDE architecture is shown in Figure 1 [Clark93a]. The various modules of the APDE will communicate with the APIB through remote access. Each module in Figure 1 represents a phase of the Application Protocol development process. The software tools making up the APDE each support one or more Application Protocol development phase. Some tools will be run remotely using the EXPRESS Server. Other tools will be client applications that will reside on PCs or workstations belonging to APDE users.

[1] An Integrated Resource (IR) is a generic specification defined in STEP which is referenced and usually specialized by multiple APs.

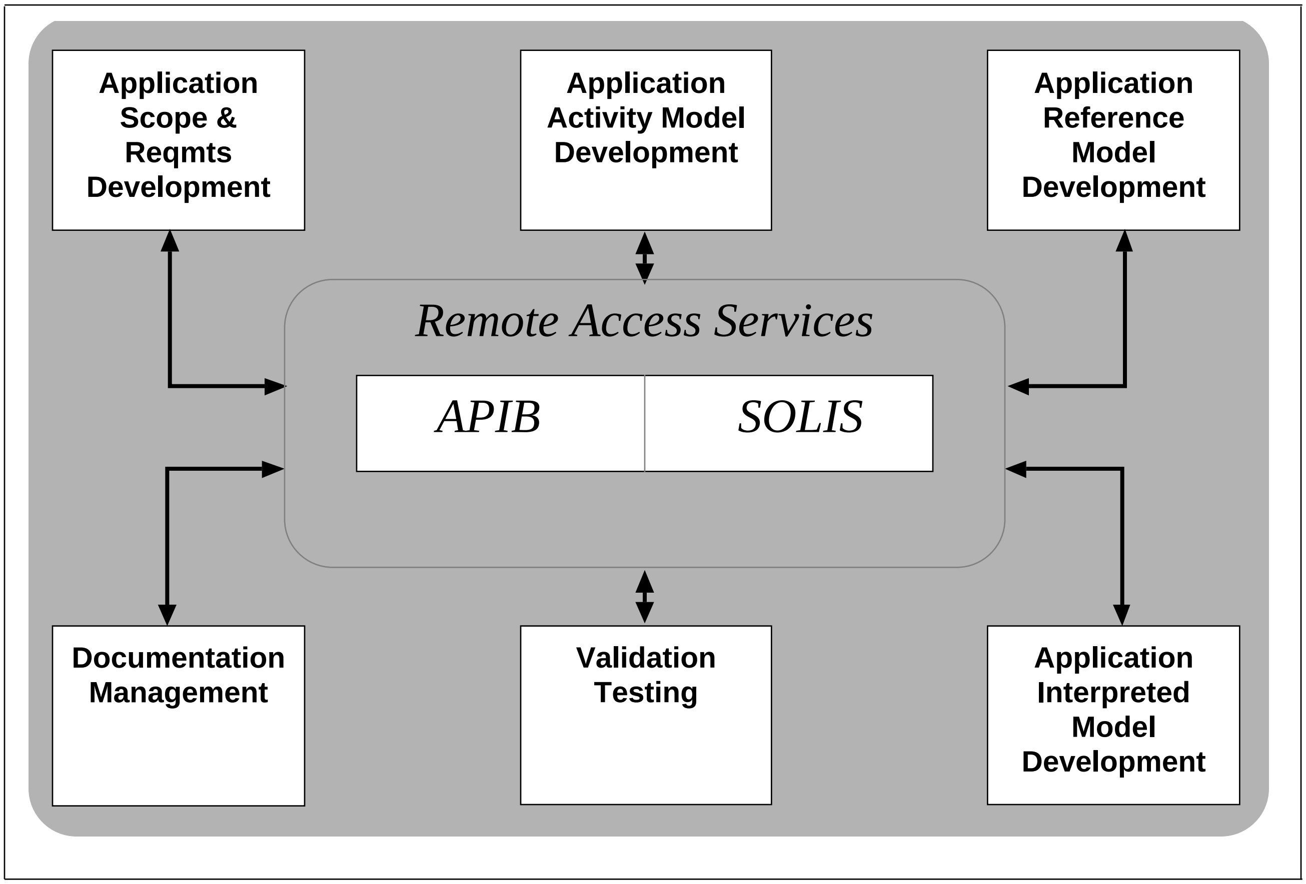

Figure 1: Future APDE Architecture

3 SGML Basics

In order to provide some of the services described in section 2, the APIB will provide a large portion of STEP documents in the SGML format. As previously mentioned, SGML is an international standard (ISO 8879) for representing the structure of a document [Goldfarb]. SGML is understood to be one of the most important formal standards in electronic publishing [Ressler]. There are two types of documents in an SGML environment: document type definitions (DTDs) and one or more instances of the DTDs as described below.

3.1 Document Type Definition

The structure for a document type is specified in a Document Type Definition (DTD). Using SGML's *element* keyword, a DTD writer specifies a context free grammar for the document's structure. Each element in the DTD specifies a content model for each region in a document. The DTD represents a tree structure of the document where each node in the tree corresponds to an element which can be further broken down into other elements. The name of an element directly follows the keyword *ELEMENT*. The content model for the element is specified within parentheses. For example, consider the following excerpt[2] from the NIST DTD for STEP Integrated Resources [Phillips]:

```
<!ELEMENT Foreword.C (  Foreword.C.H,
                        Foreword.OS,
                        Foreword.NA.Ref,
                        Foreword.IA.Ref,
                        Foreword.Bdy) >
```

Each item in the above content model is itself an element. The parent element, *Foreword.C*, represents the Foreword Clause of an Integrated Resource as specified by the Supplementary Directives for the presentation of STEP documents [Shaw]. The content model for *Foreword.C* specifies that the Foreword clause of an Integrated Resource begins with a Foreword heading, followed by standard opening sentences, followed by the references to the normative annexes, followed by references to informative annexes, followed by the body of the Foreword clause. This content model is relatively simple, however, content models can also be used to denote more complex structures such as "or" groups, optional elements, or repeating elements. Also, when the content model of an element consists of only standardized text that is to be generated (boilerplate text), the DTD can include a general entity definition which is used to specify the text for the corresponding element in the DTD. For example, because the opening sentences of the foreword section of Integrated Resources are standard, boilerplate text, an entity *Fore-*

[2] The excerpts shown may not correspond with future versions of the DTD created after publication of this document.

word.OS.Text is defined which specifies the content of the text for the corresponding element *Foreword.OS*. General entities are then referenced in the document instance as will be described in section 3.2.

DTDs may also use the *ATTLIST* keyword (meaning attribute list) to associate properties with elements. For example, an element containing a graphic would probably have an attribute referring to the file containing the graphic and another attribute specifying the graphic's format.

Thus far, a DTD for STEP Integrated Resources and a DTD for STEP Application Protocols have been developed as part of the SGML environment for STEP. These are the two primary documents for which DTDs will be required; however, we are considering other STEP documents that may also be well suited for representation in SGML.

3.2 SGML Document Instance

A text document conforming to the grammar defined in a DTD is called a *document instance* of the DTD. Because SGML is a platform and system neutral representation, document instances can be shared by users with varying hardware and text processing software. The document instance consists of the original text of the document embedded with tags that correspond to the elements defined in the DTD. The tags denote the structure of the document instance and must conform to the structure defined in the DTD. For example, the following excerpt from a STEP Integrated Resource document instance conforms with the DTD excerpt shown above:

```
<Foreword.C>
      <Foreword.C.H>Foreword</Foreword.C.H>
      <Foreword.OS>&Foreword.OS.Text;</Foreword.OS>
      <Foreword.NA.Ref>&Foreword.NA.Ref.Text;</Foreword.NA.Ref>
      <Foreword.IA.Ref>&Foreword.IA.Ref.Text;</Foreword.IARef>
      <Foreword.Bdy> This Part of ISO 10303...</Foreword.Bdy>
</Foreword.C>
```

Notice the use of SGML general entity references which are delimited by an opening ampersand sign (&) and a closing semicolon (;) such as Foreword.OS.Text above. These general entities are defined in the DTD and specify the text for the entity with which the entity references correspond. This provides the ability to insert standardized boilerplate text at specific locations in the document. For example, the entity Foreword.OS.Text is defined in the DTD as follows:

```
<!ENTITY Foreword.OS.Text

"This Part of ISO 10303 has been prepared by Subcommittee 4 (Indus-
trial data and global manufacturing programming languages) of ISO
Technical Committee 184 (Industrial automation systems and integra-
tion)" >
```

An SGML application will resolve entity references in a document instance by replacing the entity reference with the replacement text as specified in its definition in the DTD. The use of general entities in the document instance reduces the need to re-enter text that has already been defined once and helps to ensure that standard boilerplate text is accurate and complete. Further discussion of automatic generation of boilerplate text and other benefits of using SGML follows in section 4.

4 Why SGML?

As has been suggested in the previous sections, the SGML standard can provide many benefits to STEP. Because SGML is a computer-interpretable language for representing the structure of documents, SGML can improve both the STEP documentation procedures as well as the usefulness and quality of STEP documents. SGML can ease documentation procedures by providing document shareability and by providing support for structure-enforced editing and other time-saving features such as automatic generation of boilerplate text. SGML can also improve the usefulness and quality of STEP documents by increasing the integrity of information in documents, by expanding the potential for collaboration on document preparation, and by increasing document reusability. All of these benefits are described in further detail below.

4.1 Document Shareability

SGML documents are shareable because SGML documents are in a neutral, ASCII-based format. Thus, SGML documents are machine-independent and can be exchanged between users of heterogeneous computer systems. This feature alleviates the inherent problems found in using and translating between proprietary formats. These problems include data redundancy when creating multiple versions of the same document to support various formats and system configurations and loss of data content such as tables, graphics or formatting information when translating between the various formats. In an SGML environment, a single SGML version of the document could be shared by any software system.

Also, unlike proprietary formats, SGML does not represent style information such as the specification of bold or highlighted text within the SGML document. There are a number of mechanisms for storing style codes outside of the document in SGML. Some SGML-based software applications allow the user to apply style to elements in a separate specification called a style sheet. Some of these applications use proprietary style sheets. However, standard formats for specifying style sheets are currently under development. With the growing support of these standards in SGML software applications, the movement is toward the specification of style sheets which also can be shared by different applications on different platforms.

4.2 Structure-Based Editing

Currently the structure for STEP documents is enforced manually. That is, the editing committee chair or qualification project within ISO Subcommittee 4 (SC4)[3] reviews the documents manually to ensure that each Part conforms to the specifications in the Sup-

[3] This subcommittee is responsible for STEP and is part of TC184 (Industrial Automation).

plementary Directives. An SGML editing tool, however, automates the process for document structure enforcement and allows the author of the document to self-test the structure of the document as it is being edited. The user is guided through the editing process by being prompted for required or optional elements and warned of elements which are illegal or missing. This allows the author of the document to focus on the technical content while the structure-based editing tool ensures that the document structure and format are correct.

4.3 Automatic Generation of Text

An application can make use of SGML to allow boilerplate text to be automatically generated during editing or printing. This is especially useful in the STEP environment because of the large amount of rote, boilerplate text which is required in the documentation. The NIST DTD for Integrated Resources and the DTD for Application Protocols specify all boilerplate text that is required in a STEP Integrated Resource and AP respectively as set forth by the Supplementary Directives. Thus, when the Supplementary Directives change, those changes can be indicated in the DTD, and all documents conforming to the DTD will be automatically changed accordingly. This feature assures that generated text is consistent with the Supplementary Directives and with other Integrated Resources and APs which were developed using the STEP DTDs.

4.4 Information Integrity

Much of the content in STEP documents uses or refers to information in other STEP documents or to information in other areas of the same document. However, because the standard is still evolving, referenced information is often unstable. When a referenced document or part of a document is changed, the referencing document must also be changed accordingly. Currently, this process is done manually which makes it difficult to ensure the integrity and consistency between and across the Parts of the standard. This issue can be addressed by employing the cross-referencing capability enabled through the use of SGML tagging. Using the tags in an SGML document as cross-reference points, the user can define cross-references between or within documents. Therefore, when a change in one document occurs, the other document can be updated automatically. When maintaining the interdependencies between STEP Parts[4], this capability helps to assure that the information referenced is always current and accurate.

4.5 “Intelligent” Documents

Unlike many text formatting languages, SGML represents content information that can be used in a number of ways. For example, a browsing tool can use the SGML markup

[4] See section 8 for a definition of “STEP Part”.

in a document instance to determine which regions in the document should be displayed and to determine the anchor points for hypertext links in a hypertext-based text browsing system such as World Wide Web.

Also, because SGML documents can be "intelligently" processed by a computer, SGML documents can easily be stored in a DBMS. We will include an SGML-based DBMS in the aforementioned APIB to store a large number of STEP documents. This will provide additional support, such as automated version and access control capability, to facilitate multi-user read and/or write access to STEP documents. It will also allow linking to other STEP-related documents in the APIB such as issue logs and guidelines documents.

SGML documents can also be used by structure-based querying applications. For example, when performing searches on SGML data, the user can limit the scope of the search based on the structural elements defined in the DTD. This allows the user to quickly get to the particular information within a document that is needed which is particularly useful in STEP due to the size and complexity of STEP documents.

4.6 Increased Potential for Collaboration

The average size of a STEP Part development team is roughly four to five members all of whom are usually spread geographically across the world. Because of the high level of interdependence among the Parts in STEP, these geographically dispersed teams must frequently interact and work with each other. However, due again to the size and complexity of STEP documents and largely to the use of proprietary formats, it is difficult for teams and team members to coordinate their efforts, particularly when trying to transfer, print or update STEP documents.

An SGML environment, however, would help to alleviate these problems by providing a neutral environment in which to exchange STEP documents. By employing a set of STEP-specific SGML DTDs and their corresponding style sheets instead of proprietary formats to create STEP documents, editors could reduce the document transfer, print and revision problems inherent in using proprietary formats. This would effectively increase collaboration among STEP document editors and ensure that STEP documents developed in a collaborative effort are always consistent and structurally correct.

4.7 Reduction in Document Preparation Time

Finally, all of the above features and the single, neutral, structure-based format of SGML could significantly reduce the time needed to prepare a STEP document in SGML. The production time for ISO publication of a STEP document is roughly six months. We expect, however, that using SGML to author STEP documents during the publication process would reduce production time for publication possibly by several months which ultimately would reduce the time that valuable technical experts spend on formatting final documents.

The overall benefits of using SGML are evidenced by an increasing number of vendors which support SGML and a movement towards state-of-the-art SGML products. The size and complexity of STEP documents provide even further justification for the use of SGML for its potential STEP-specific documentation benefits.

5 The SGML Environment for STEP

In order to take advantage of the power of SGML, the applications in an SGML environment must be "SGML-aware." This means that the applications must be able to understand and interpret the semantics of the SGML elements in a DTD and corresponding tags in a document instance. There are various commercial and public-domain software tools available that provide SGML awareness. Currently, NIST is developing and acquiring a subset of these tools. However, the APDE will eventually provide an entire, self-contained SGML environment. In this section, we discuss the SGML software tools that make up the components of a complete SGML environment for STEP, including essential and additional nonessential components.

Figure 2 provides a pictorial representation of the component tools in the SGML environment for STEP. The rectangular boxes represent stand-alone tools. Ovals represent tools or functionality that are typically contained within a stand-alone tool. Stacked rectangles represent documents. The disk represents the repository of documents managed within the SGML environment.

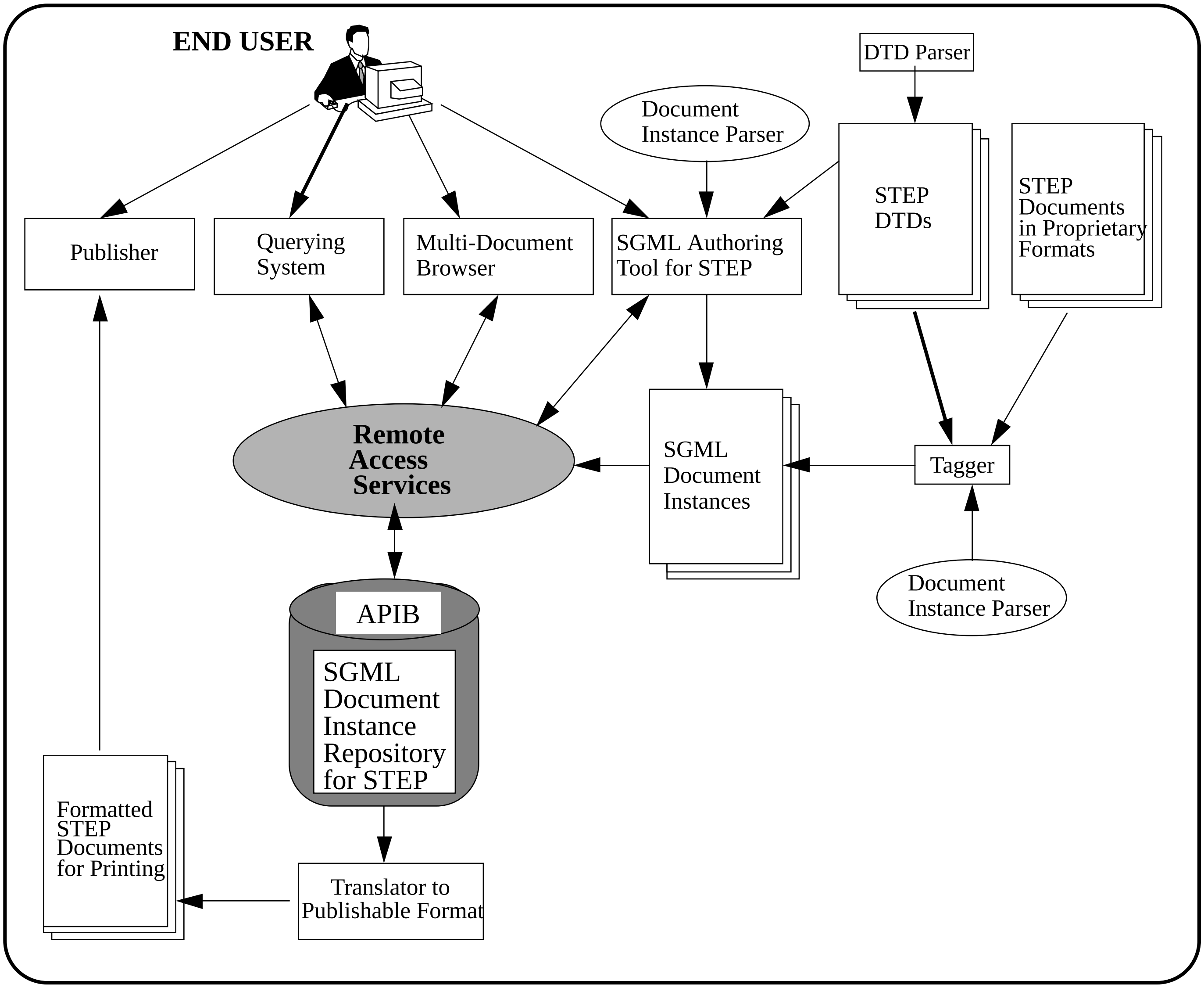

Figure 2: The SGML Environment for STEP

5.1 Essential Components

The essential components in an SGML environment for STEP include a DTD parser, a document instance parser, a STEP-customized SGML authoring tool, a tool for converting documents from proprietary formats into SGML, and a translator from SGML to a publishable format. These components facilitate basic SGML functions which include creating, editing and printing STEP documents in SGML.

5.1.1 DTD Parser

The DTD parser is used to check the syntax of the DTD to ensure that the DTD conforms to the proper SGML syntax. Due to the lack of an official conformance testing system for SGML, the behavior of the available DTD parsers and the type of errors that each of them reports vary. Thus, it is often useful to validate a DTD against several parsers to fully ensure the quality and syntactical correctness of the DTD.

A DTD should also be validated against a document instance that is representative of the class of documents for which the DTD is modelled. This is done using a document instance parser, as described below. In case of a discrepancy between the DTD and the document instance, both should be checked to see where the discrepancy has occurred. In some cases, it is discovered that the DTD should be revised to better reflect the structure of the corresponding document instances.

NIST is creating, validating and maintaining DTDs for STEP based on requirements from the STEP editing committee. Therefore in order to validate thoroughly the STEP DTDs, the APDE includes multiple DTD parsers as part of the SGML environment for STEP.

5.1.2 Document Instance Parser

Each document instance must be validated against a DTD. This is done using the document instance parser which checks to see if the tags and text in a document instance comply with the elements defined in the corresponding DTD.

An instance parser can be stand-alone, but is usually contained as part of an SGML authoring tool or SGML conversion tool. When an instance parser is included in an authoring or conversion tool, it is used to validate a document instance as the document instance is being edited or converted, respectively. The SGML environment for STEP will include an instance parser which will be provided as part of an SGML authoring tool customized for STEP.

5.1.3 STEP-Customized SGML Authoring Tool

An SGML authoring tool allows the user to create and edit SGML documents. With a generic SGML editor, the user can perform basic SGML editing functions which include insertion, deletion, cutting and pasting of text and SGML tags. A generic SGML authoring tool helps the editor to insert text and tags in the document instance so that the inserted tags correspond with the ordering and the placement of the elements defined in the DTD. The user can validate the document instance against the DTD using an internal document instance parser.

In addition to the above basic SGML functions, functionality specific to the STEP DTDs should also be provided by an editor in a STEP SGML environment. For example, in order to free the user from manually re-entering boilerplate text (see section 4.3), the DTDs for Integrated Resources and APs were modelled such that most boilerplate text can be generated by an application.The customized editor must, therefore, be programmed to provide this text so that the user is able to view generated text while editing.

Also, the customized editor should provide DTD-specific browsing so that the user can interactively select which region or regions in the document will be viewed while editing. Furthermore, a customized editor should include a printing capability which allows the user to convert the document to a publishable format (see section 5.1.5). These additional features will help to provide the true usefulness of SGML by using the structures defined in the DTDs to tailor an SGML authoring tool for editing STEP documents in particular. Like the other tools in the SGML environment, the customized editor will be integrated with the other components of the APDE so that the editing of STEP documents is tightly coupled with the other phases of the AP development process.

5.1.4 Conversion of Legacy Documents to SGML

Because all of the Parts of STEP are currently produced in LaTeX or WordPerfect, the SGML environment will include (at least in its initial stages) a mechanism for converting documents in proprietary formats into SGML. We have identified the following three tools which could be used toward this end: 1) a dedicated automatic conversion tool; 2) an SGML-aware editor; 3) or an ASCII text editor. A dedicated automatic conversion tool would allow the user to convert a document of proprietary format such as WordPerfect into SGML by applying a set of conversion rules based on the text and style codes in the original document. A conversion tool usually includes an embedded document instance parser for parsing documents against a corresponding DTD as they are converted to SGML.

An SGML editor could also be used for converting legacy documents into SGML. In order to do this, the user would first generate an ASCII version of the document and then read it into an editor and insert the appropriate SGML tags. An SGML editor would typically allow the user to select from a list of “legal” tags based on the DTD and, using an internal instance parser, the document could be parsed as it is being converted.

A significantly less satisfactory solution would be to use a non-SGML ASCII text editor such as Emacs or vi. However, this would require the user to manually insert tags and then use an external document instance parser to parse the document against a DTD. This process would be error prone and would require a thorough understanding of SGML and the DTD.

While the SGML editor is in some ways preferable to a non-SGML editor, both of these methods have their disadvantages. A major disadvantage of both methods is that when converting legacy documents into ASCII, some data, such as graphics and tables, are not carried over into the ASCII representation of the document. Also these methods are more time-consuming, especially when dealing with large and complex documents. Another disadvantage is the lack of reusability of these methods. Unlike using a dedicated SGML conversion tool, using an editor would not allow the user to apply and reapply a set of rules to several documents of the same format. Thus each document is converted without the time-saving benefit of using procedures applied to documents previously converted.

A dedicated SGML conversion tool is the goal for converting proprietary documents into SGML, primarily because this method would provide SGML-awareness as well as procedure reusability. We are now in the process of acquiring an SGML conversion tool for which we will need to develop autotagging scripts for our DTDs in particular. The interim solution in the APDE SGML environment has been to use an SGML editor to convert legacy documents into SGML. With this editor, several existing STEP Integrated Resources have already been converted using an abbreviated DTD for demonstration purposes, and three Integrated Resources have been converted with a full, unabbreviated DTD. As we move toward the use of the automated SGML conversion tool, we will provide regular feedback to the vendor so that we will be able to provide a more efficient SGML conversion service in the near future.

5.1.5 Translation From SGML to a Publishable Format

In order to publish an SGML document instance, the document instance must first be translated into a format recognizable to publishing software. This can be accomplished by using a translator which converts an SGML document instance into a publishable format using a set of rules based on the DTD and the markup in the corresponding document instance. The APDE will support translations of SGML document instances to LaTeX. LaTeX was chosen as the publishing medium because LaTeX is one of the currently acceptable electronic representations for STEP documents, and LaTeX templates have already been written for Application Protocols and Integrated Resources. Also, LaTeX is in the public-domain, and public-domain previewing and printing software is available for Unix workstations, PCs, and Macintosh computers.

Although there is no third-party SGML translator that directly supports STEP DTDs, there are some SGML tools available that make it easier to write such a translator. These tools provide a scripting language that is used to specify translation instructions for each element, attribute value, or other SGML constructs found in a document instance. Other tools provide a framework for developing translators but do not directly support SGML. We are currently evaluating third-party software to help us write custom translators for STEP from SGML into LaTeX.

5.2 Additional Components

In order to reap many of the more advanced benefits of SGML, the following components should also be included in an SGML environment. These components are not essential, but would provide valuable, additional functionality.

5.2.1 Multi-Document Queries and Browsing

The APDE's SGML environment will support queries and browsing across more than one document. When performing operations involving SGML documents, the user can limit the scope to regions defined in the DTD. Because STEP documents are so large, this filtering ability will be an important feature of the SGML environment. As an example of how filtering can be useful in browsing STEP AP documents, suppose that the DTD for APs specifies an element *schema* for representing EXPRESS schemas with an attribute *name* for the schema name. A query processor could be implemented that would allow users to restrict their searches to schema information within APs or even to a particular schema. Similarly, an SGML browser could allow a user to view the names of schemas in APs, taking advantage of the document structure information encoded in the AP document instances.

In addition to using SGML markup as a means of information hiding, the SGML environment will use SGML as a vehicle for specifying hypertext links. This will be accomplished by using SGML attributes to specify references from one element to another. These references can either be from one element to another within the same document instance or they may be inter-document hyperlinks. An example of where these hypertext links would be useful is near the beginning of AP and IR documents where a collection of terms are listed in groups according to the STEP Part where they are defined. Using hyperlinks, a browser could bring up the actual definition of the term from the Part in which the term is defined, saving the user the trouble of having to browse through another document in order to find the term's definition. Hyperlinks in the SGML environment may be provided using HyTime [Newcomb], an emerging international standard which specifies a method of implementing SGML cross-referencing and other useful SGML-enabled capabilities. Because HyTime is relatively young, few third-party SGML tools currently support HyTime. However, we expect this situation to change in the future as more people develop SGML-based hypermedia applications. Therefore, we are considering using HyTime as part of our future plans.

The APIB will also include some SGML database functionality to support multi-document querying and browsing. This database will include an indexed repository of SGML documents which will support efficient retrieval based on SGML structure. It will also support full-text retrieval for non-SGML-structured searches. Additionally, the database will have an open architecture and an application programmer's interface so that client software for browsing and querying can be easily written and seamlessly integrated into the APDE.

We have investigated commercial SGML-aware databases for use in the APDE. Although many vendors claim some support for SGML, the extent to which their products are tailored for SGML applications varies widely. Also, most SGML database tools are expensive. Nevertheless, we have recently selected a commercial text database product which we expect will be cost-effective for the APDE. This product, unlike many other text retrieval products, can index texts according to regions corresponding to SGML elements. It also has an open architecture that facilitates its integration with other APDE components.

5.2.2 Remote Access Services

As mentioned earlier, the APIB will provide remote access services to APDE users. Thus the APIB must support concurrent read-only access and single user write access for SGML documents. Access privileges will be granted with respect to a privilege list which will be stored in the APIB. The privilege list itself could possibly be represented using SGML.

Configuration management (CM) must be provided in conjunction with these remote access services. APDE users will have a client CM application that they will use when they wish to modify STEP-related documents or view a document’s version history. This CM application will provide functionality similar to that of the Revision Control System (RCS), a public-domain CM tool used for software development at NIST [Bodarky91]. In addition to the client CM applications, there will be a CM application on the APIB server at NIST that will keep track of access rights for all APDE users and version information for all APIB documents. Like the access privilege information, documents’ version histories could be represented in SGML.

By using the document structure defined in the STEP DTDs, the SGML environment could provide remote access and CM services with a finer degree of granularity than a typical file-based system. For example, the SGML environment may support two users simultaneously editing different sections of the same AP document. This would be accomplished by means of granting write access for document regions defined by SGML elements.

Another area in which SGML will add functionality to remote access services is in measuring usage statistics. Because documents are represented in SGML, the SGML environment will track which regions in STEP documents are accessed the most frequently and which regions undergo the most modification. The usage statistics themselves could be stored in a special SGML document. The information gleaned from these usage statistics can then be used to gain additional insight about the AP development and AP implementation processes.

The database software we are planning on using in the SGML environment (see section 5.2.1) has a client-server architecture and supports remote access. The software includes a customizable query interface and document viewer interface, both of which can be used to access a database server over a wide area network.

We are also considering applying the technology used in World Wide Web (WWW) and NCSA Mosaic, a public-domain WWW browser, to the APDE. WWW documents are represented using HTML, a subset of SGML that supports hypertext browsing. By translating STEP documents into HTML documents, we can make them accessible on the Web. However, because HTML is more restrictive than SGML, the STEP documents would lose a lot of their SGML-defined structure in the translation.

5.2.3 Multi-User Editing

Multi-user editing would provide a useful feature to STEP developers because most STEP documents are produced by committees. A multi-user editing environment would help to address some of the difficulties found in several users using separate editors to edit the same document by providing configuration management and remote access services to monitor and control user access and track revisions made to documents. Also, multi-user editing would reduce both the number of copies of documents and the number of conversions to various proprietary formats.

Multi-user editing will most likely be provided in the APDE SGML environment as part of a text-based SGML-aware DBMS system. Editing privileges could be granted on a per Part or per region of a Part basis using document identifiers and the elements defined in the DTDs as areas of write accessibility.

6 Conclusion

In this paper, we have given an overview of an architecture for an SGML environment for STEP. As part of the APDE project, implementation of this SGML environment is now underway. In developing the SGML environment for STEP we are encountering some challenges in integrating the available commercial and public-domain SGML software. We have found that while commercial SGML software tends to be more advanced and easier to use than public-domain software, commercial SGML software is typically expensive and often difficult to integrate with other software modules. On the other hand, public-domain SGML tools are often well suited for integrating with other software; however, they are often not as easy to use as some commercial tools. Thus there is a trade-off between ease of use and integrability that must be carefully balanced.

Another issue is that there is a large variance in the degree to which SGML products, both commercial and public-domain, conform to the SGML standard. It is common for SGML users to have trouble moving their DTDs and document instances from one environment to another because of lack of conformance to the standard by at least one of the SGML software products. While there are work-arounds to this problem (modifying DTDs and/or document instances according to the particular environment in which they are being used), it is our goal to use products in the SGML environment for STEP that conform to the SGML standard and that would thus provide the portability and shareability of SGML documents for which the use of SGML is intended.

We are optimistic that third-party SGML products will have far fewer limitations in the future as SGML becomes more popular and the SGML market becomes economically viable for vendors to support. Also, new and better public-domain tools continue to become available. Furthermore, nonconformance issues are currently being addressed in the SGML user community, and work is underway at NIST within the Computer Systems Laboratory to establish conformance testing criteria and a conformance testing program for SGML software [Wilson].

6.1 Status of the SGML Environment and Near-Term Plans

Thus far, we have written the DTDs for STEP IRs and APs, and we have begun to tag documents using those DTDs. We have also developed a prototype SGML authoring environment for STEP and integrated it with the APDE demonstration system. Our future plans for the near term focus on three objectives: increasing the functionality of the authoring environment, providing printing capability, and building a document repository with browsing and querying services. We have identified a number of pieces of commercial and public-domain software to help us achieve these goals. In particular, we have identified and are in the process of acquiring tools for building a customized SGML editor, a tool for implementing translators between SGML and publishable formats, and database software.

6.2 Long-Term Plans for the SGML Environment

Our longer-term plans include implementing the SGML repository on a server at NIST and making it remotely accessible to STEP AP developers. We are also planning on using SGML to help implement configuration management as was discussed earlier. As the APDE's SGML environment becomes a client/server system, some interesting issues will arise concerning which services will be provided locally versus remotely. Our approach may be to provide third-party products remotely, but to provide NIST-developed and public-domain software directly to the user. Another issue is that AP developers use both DOS-based and UNIX-based platforms, thus SGML client applications must be available to support AP developers in both of these environments.

These and other issues will be resolved as the SGML environment is planned and developed through iteratively collecting APDE user requirements from the STEP community. These requirements will be used to further solidify and refine the implementation plan for the SGML environment and to guide its future development. We also plan to work with one or two AP development projects, allowing them to use and test the various products in the SGML environment as they are being developed and refined.

When we have reached a complete workable environment, we will extend the SGML environment to the entire STEP community, providing them with user training and other tools, such as automated SGML conversion tools, which are necessary to effectively introduce SGML into the STEP development environment. We will also continue to seek feedback from the user community and to enhance the SGML environment accordingly.

While our goals for the SGML environment are ambitious, we believe that these goals can be accomplished as we continue to gain experience as SGML implementors and as the tools and technology available to us continue to improve.

7 References

[Bodarky91] Bodarky, S., *A Guide to Configuration Management and the Revision Control System for Testbed Users*, NISTIR 4646, National Institute of Standards and Technology, Gaithersburg MD, August 1991.

[Bodarky93] Bodarky, S., Paisley, S., *An SGML DTD for the STEP Integrated Resource Parts*, NISTIR 5224, National Institute of Standards and Technology, Gaithersburg MD, July 1993.

[Clark93a] Clark, S., Barnard Feeney, A., *APDE Demonstration System Architecture*, NISTIR 5318, National Institute of Standards and Technology, Gaithersburg MD, January 1993.

[Clark93b] Clark, S., Barnard Feeney, A., Fowler, J., *Specifications for an Application Protocol Development Environment*, NISTIR 5248, National Institute of Standards and Technology, Gaithersburg. Md., August 1993.

[Goldfarb] Goldfarb, C., *The SGML Handbook*, Oxford University Press, New York, 1992.

[ISO] ISO 10303-1:1994 *Industrial automation systems and integration—Product data representation and exchange—Part 1: Overview and fundamental principles.*

[Libes] Libes, D., *Concepts of the NIST EXPRESS Server,* Proceedings of the First International Workshop on Services in Distributed and Networked Environments (SDNE), Prague, Czech Republic, June 27-28 1994.

[Newcomb] Newcomb, S., Kipp, N., Newcomb, V., *The Hytime Hypermedia/Time-based Document Structuring Language*, Communications of the ACM, Vol. 34, No. 11, November 1991.

[Mitchell] Mitchell, M., *Development Plan Validation Testing System*, NISTIR 4417, National Institute of Standards and Technology, Gaithersburg MD, September 1990.

[Phillips] Phillips, L., *A DTD for STEP Integrated Resources.* ISO TC184/SC4 Editing Committee Version 1.0 N43, September 5, 1994.

[Ressler] Ressler, S., *Perspectives on Electronic Publishing, Standards, Solutions, and More*, PTR Prentice Hall, NJ, 1993.

[Rinaudot] Rinaudot, G., *The STEP On-Line Information Service*, DRAFT NISTIR, National Institute of Standards and Technology, Gaithersburg MD, May 1994.

[Shaw] Shaw, N., *Supplementary Directives for the Drafting and Presentation of ISO 10303,* Version 1.1, July 10, 1993.

[Wilson] Wilson, R., *SGML Parser Validation Procedures*, to be published as a NISTIR, National Institute of Standards and Technology, Gaithersburg MD, 1994.

8 Glossary

Application Protocol (AP)
A Part in STEP which combines components of Integrated Resources, selects implementation mechanisms, and uses the description methods to specify product data to be exchanged and the meaning of that data in a particular industrial context [Clark93b].

Application Protocol Development Environment (APDE)
An emerging software environment whose goal is to facilitate the accelerated development of high quality STEP Application Protocols through providing an integrated set of tools tailored to the needs of STEP.

Application Protocol Information Base (APIB)
A repository of STEP documents and data, including STEP Application Protocols, Integrated Resources, and other documents and data models related to STEP, which will provide DBMS capability such as query look-up, multi-user access, version control and configuration management.

Boilerplate Text
Standard text specified in the Supplementary Directives which is to be included in every document in a given document class.

Data Model
A description of the information requirements, relationships between informational objects, information structure, and constraints for a subject area.[Mitchell].

Document Class
A group of documents which have a uniform purpose, structure or format and which can be modeled collectively in a single SGML DTD.

Document Instance
An SGML document which contains tags and other SGML constructs which conform to a given SGML DTD.

Document Type Definition (DTD)
A formal computer-interpretable specification in SGML which describes the structure and semantics of documents in a document class.

Integrated Resource (IR)
A Part in STEP which provides information models describing generic constructs which are useful in a wide variety of product descriptions, including Application Protocols and other Integrated Resources.

SGML-aware
A tool or software system that uses the structure defined in a DTD and document instance to process or evaluate data.

SGML element

An SGML construct defined in the DTD which describes a particular region in a document. Each element may contain other elements. The collection of elements in a DTD comprises a tree structure which describes the overall structure of a document.

SGML entity

An SGML construct defined and used in the DTD or document instance which specifies a string of replacement text to be used when the entity is referenced.

SGML Tag

An SGML construct used in document instances which delimits a particular region in a document instance and corresponds with an element defined in the DTD.

Standard for the Exchange of Product Model Data (STEP)

An emerging ISO standard (ISO 10303) which specifies the requirements for the exchange of manufacturing data in multiple industrial contexts across heterogeneous computer systems.

Standard Generalized Markup Language (SGML)

An ISO standard (ISO 8879) which is used to specify the structure of a document.

STEP Part

STEP consists of a collection of specifications. Each individual specification is known as a “Part” of STEP. Major classes of Parts in STEP include APs, IRs, and categories not discussed in this paper such as Description Methods, Implementation Methods, and Conformance Testing procedures. See [ISO] for an overview of the STEP Parts.

Appendix A: The STEP DTDs

In order to provide some of the services described in Section 2, the APIB will represent a large portion of its contents in SGML. A DTD for STEP Integrated Resources and a DTD for STEP Application Protocols have been developed as part of the SGML environment for STEP [Phillips]. These are the two primary document types for which DTDs will be required, however, we are considering other STEP documents that may also be well-suited for representation in SGML. The structure defined in these DTDs was specified according to the Supplementary Directives for the presentation of STEP documents [Shaw]. Therefore, the use of these DTDs will help to ensure the structural correctness and consistency of STEP documents. The STEP DTDs are available to the public electronically, and we encourage others to use them to build SGML-aware applications for STEP.

Prior to creating these DTDs, several existing DTDs were examined to determine whether they were applicable to STEP. Two of these existing DTDs were ISO 9573-11: Application at ISO Central Secretariat for International Standards and Technical Reports - First Edition and MIL-M-28001-A: Markup Requirements and Generic Style Specification for Electronic Printed Output and Exchange of Text. Both of these DTDs were determined to be unsuitable for STEP because they make assumptions about document structure that are invalid for STEP and because, due to their being written to cover a wide range of documents, they are too complex. Therefore, it was determined that STEP required its own specialized DTDs [Bodarky93].

Creating the STEP DTDs turned out to have the beneficial side effect of uncovering several ambiguities and inconsistencies in the STEP Supplementary Directives. It was also found that some existing STEP documents violated the Supplementary Directives. These conflicts and ambiguities illustrate the difficulties in producing highly specialized documents while relying on rules written in English and interpreted by humans [Bodarky93]. In contrast, the STEP DTDs provide a formal specification for Application Protocols and Integrated Resources and are not susceptible to human misinterpretation.